

REMARKS

This is in response to the Office Action mailed 2/15/2008 and further in view of the Examiner Interview of 6/10/2008.

Applicants are appreciative of the opportunity for an interview and the professional and courteous manner in which the interview was conducted. During the interview, a proposed amendment was discussed with respect to the amendments submitted with the current amendment. Applicants emphasized that these amendments are an attempt to clarify the invention without adding new matter. During the interview of 6/10/2008, Examiner Farhan reviewed the proposed amendment and commented that it did appear to move the prosecution forward.

Reconsideration of this application is respectfully requested in view of this response/amendment.

STATUS OF CLAIMS

Claims 27-41 are pending.

Claim 1-26 were previously cancelled.

Claims 27, 33 and 39 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement.

Claims 27-41 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the article entitled, “An Efficient XML Schema Typing System,” to Wang et al., hereafter “Wang”,

in view of the article entitled, “Extending Tree Automata to Model XML Validation Under Element and Attribute Constraints,” to Bouchou et al., hereafter “Bouchou”.

OVERVIEW OF CLAIMED INVENTION

The presently claimed invention provides for a computer-based method for compiling a structured document schema (e.g., XML schema) into type annotation records. In an initial step, a type hierarchy is built from an XML schema, based on a derivation of relationships among types. For each complex type in a schema, a type record is created. Each type record in a type hierarchy contains typing tuples for sub-elements and attributes of a specified type. Assuming no overlap, both element and attribute names are listed together, in a type record. If there is an overlap between an element and attribute name, a specified string is prefixed to an attribute name. Each typing tuple is comprised of a type name, element name, or string-valued attribute name as a first field, a type identifier as a second field, and a parent element name as a third field. If an element is of global element type, the corresponding third field will remain empty. For each type record, all tuples are determined in this manner. After all tuples are determined, a typing set is formed by the union of all typing tuples corresponding to type records formed. The number of typing tuples in a type record is dependent on the number of sub-elements and attributes for a given element type. Next, a typing set is sorted with respect to a first string field, in alphabetical order. Next, an ambiguity typing sequence is created for those tuples sharing a common first field and having a unique second field. Third fields from typing tuples in an ambiguity typing sequence are then collected and sorted. Since it is necessary for global types to be unique, a collection of third fields from an ambiguity typing sequence should not contain any empty members. After third fields are sorted, an offset number is assigned to each typing tuple in

accordance with its position in sorted order. As a next step, typing tuples within each ambiguity typing sequence are then arranged based on the unique offset numbers assigned to each third field. Each offset number assigned to each third field is unique within an ambiguity typing sequence since there is no ambiguity within each parent element. Following the step of sorting and arranging, a type array is created by extracting types found in the second field of a typing tuple according to the sorted order of ambiguity typing sequences. Types not included in ambiguity sequences, which are also extracted from the second field of typing tuples, are listed following those typing tuples that are members of an ambiguity typing sequence. It is of note that multiple entries for a given type may exist if the type is included in multiple ambiguity sequences. Entries in a type array that correspond to type names with an offset number assigned as described previously, are also given the same offset number. Those entries that have no offset number are assigned an offset number of zero. As a last step in the algorithm of the present invention, an index structure is created to link each type name extracted from a first field of a typing tuple to its corresponding type. Index entries will have a string field denoting element type, a flag field denoting ambiguity, and an index field denoting the index of an element type in a type array. A flag field is given a value of ‘Y’ if a corresponding element type is ambiguous and ‘N’ if it is not ambiguous. An index field is given a value corresponding to the index of an element type in a type array if a corresponding flag field is set to ‘N’ and the first index entry in a type array for an ambiguity sequence if a corresponding flag field is set to ‘Y’. An index structure is implemented by, but is not limited to, one of the following data structures: hash tables, binary trees, and B+ trees.

REJECTIONS UNDER 35 U.S.C. § 112, 1st ¶

Claims 27, 33 and 39 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Appropriate amendments have been made to independent claims 27, 33 and 39 to clarify Applicants' invention without adding new matter. These clarifying amendments were discussed with Examiner Farhan during the interview of 6/10/2008, where it was emphasized that the clarifying amendments overcame the pending 35 U.S.C. §112 1st ¶ rejection. Therefore, Applicants respectfully request the Examiner to withdraw the 35 U.S.C. §112 1st ¶ rejection with regards to 27, 33, and 39.

REJECTIONS UNDER 35 U.S.C. § 103(a)

Claims 27-41 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the article entitled, "An Efficient XML Schema Typing System," to Wang et al., hereafter "Wang", in view of the article entitled, "Extending Tree Automata to Model XML Validation Under Element and Attribute Constraints," to Bouchou et al., hereafter "Bouchou". The rejection with respect to claims 29, 32, 35, 37, 38, and 40 are considered moot in view of their cancellation via the current amendment. To be properly rejected under 35 U.S.C. §103(a), the cited combination of references (i.e., Wang and Bouchou) need to teach each and every feature of the rejected claims. Applicants respectfully submit, as was emphasized during the interview of 6/10/2008, that the cited combination of references (i.e., Wang and Bouchou) fails to teach or suggest many of the features of the rejected claims.

Wang teaches a XML typing module that validates an XML document fragment against an XML schema and annotates the validated XML document with type information. Wang

teaches a specific example of utilizing the XML typing module is used as part of the runtime environment for XML-related languages that use an XML Schema as a type system.

Bouchou provides a method to model XML validation under element and attribute constraints. Specifically, Bouchou teaches a new formalism that deals with both element and attribute constraints. The proposed formalism involves extending the regular tree automata that allows the construction of a deterministic automaton having the same expression power of as that of a DTT.

By stark contrast, Applicants claim 27 teaches a computer-based method for compiling a structured document schema into type annotation records comprising steps of: (a) building a type hierarchy ordered tree from structured document based on a derivation of relations among types in said structured document and determining one or more tuples for each type record in said structured document; (b) forming a complete typing set of said tuples; (c) sorting said typing set by their first field; (d) creating, from sorted tuples in (c), ambiguity typing sequences for tuples having a common first field and having a unique second field, collecting and sorting a third field from ambiguity typing sequences, assigning a unique offset number to each sorted third field, and arranging said ambiguity typing sequences based on offset numbers; (e) creating a typing array by concatenating typing tuples in resulting ambiguity typing sequences of (d); (f) for each type record node, N, in created typing array, if the intersection of a set of tuples in N with any ambiguity typing sequence is not empty, then replacing first typing tuple in N by typing tuple having offset, wherein offset represents a position of an ambiguity type in a given

ambiguity typing sequence; (g) creating a type indexing data structure and indicating ambiguity type for each type name; and (h) outputting said created index structure.

The Examiner relies on Section 3.5.1 of Wang as teaching the feature of creating ambiguity typing sequences for tuples having a common first field. Section 3.5.1 merely teaches an improved data structure for efficient type look up. Specifically, Wang teaches the extension of the terminal dictionary by allowing the terminal dictionary to have the following four fields: *terminal*, *tok*, *ntypes*, and *type#*. The extended terminal dictionary provides the required efficient type lookup. However, it should be noted that extension of the terminal dictionary by no means anticipates or renders obvious Applicants features of creating, from sorted tuples, ambiguity typing sequences for tuples having a common first field and having a unique second field, collecting and sorting a third field from ambiguity typing sequences, assigning a unique offset number to each sorted third field, and arranging said ambiguity typing sequences based on offset numbers.

The Examiner also relies on the same citation as teaching Applicants feature, replacing first typing tuple in N by typing tuple having offset if the intersection of a set of tuples in N with any ambiguity typing sequence is not empty, wherein offset represents a position of an ambiguity type in a given ambiguity typing sequence. Applicants respectfully point out that the Examiner's citation and the entire Wang reference fail to teach or suggest any checking step for identifying if the intersection of a set of tuples in N with any typing ambiguity sequence is empty. Applicants also respectfully point out that the Wang reference also fails to teach or suggest the step of replacing a tuple with a tuple offset based on such a checking step.

As mentioned above, Bouchou provides a method to model XML validation under element and attribute constraints. However, Bouchou fails to remedy the failure of the Wang reference to teach or suggest many of the features of Applicants' pending claim 27. Therefore, for at least the reasons set forth above and as outlined during the interview of 6/10/2008, Applicants respectfully assert that the combination of Wang and Bouchou fail to teach or suggest many of the features of Applicants' pending claim 27. Hence, Applicants respectfully request the Examiner to withdraw the 35 U.S.C. §103 rejection with regards to Applicants' pending claim 27, and further respectfully requests allowance thereof.

The above-mentioned arguments with respect to claim 27 substantially apply to independent claims 33 and 39 as they recite many similar features as independent claim 27. Therefore, Applicants respectfully request the Examiner to withdraw the 35 U.S.C. §103 rejection with regards to Applicants' pending claims 33 and 39, and further respectfully requests allowance thereof.

The above-mentioned arguments with regards to independent claims 27, 33, and 29 substantially apply to dependent claims 28, 30, 31, 34, 36, 39, and 41 as they inherit all the features of the claim from which they depend. Therefore, Applicants respectfully request the Examiner to withdraw the 35 U.S.C. §103 rejection with regards to Applicants' pending dependent claims 28, 30, 31, 34, 36, 39, and 41, and further respectfully requests allowance thereof.

SUMMARY

As has been detailed above, none of the references, cited or applied, provide for the specific claimed details of Applicants' presently claimed invention, nor renders them obvious. It is believed that this case is in condition for allowance and reconsideration thereof and early issuance is respectfully requested.

This response is being filed with a fee for an extension of time. The Commissioner is hereby authorized to charge any deficiencies in the fees provided to Deposit Account No. 09-0460.

If it is felt that an interview would expedite prosecution of this application, please do not hesitate to contact Applicants' representative at the below number.

Respectfully submitted,

/ramraj soundararajan/

Ramraj Soundararajan
Registration No. 53,832

IP Authority, LLC.
4821A Eisenhower Ave
Alexandria, VA 22304
(703) 461-7060

August 5, 2008